EXHIBIT A

AMD-90001

Zorak IO Hub

Data Sheet

Publication Issue Date:	Rev:

Trademarks

AMD, the AMD logo, AMD Athlon, and combinations thereof, AMD-751 and AMD-765 and AMD900 are trademarks of Advanced Micro Devices, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

1 Overview

Revision History

Revision	Date	Comments	Author
0.1	6/16/00	Initial Release	L. Hewitt/ F. Barth
0.2	7/11/00	- LDT clock 200 MHz only	J Winkler
0.4	9/29/00	- change part name to AMD90001 - change package size to 492 - update to sync with OPUS0.26 - update revision number to global ZORAK MAS number	F. Barth

1.1 Features

The AMD Athlon™ processor powers the next generation in computing platforms, delivering the ultimate performance for cutting-edge applications and an unprecedented computing experience.

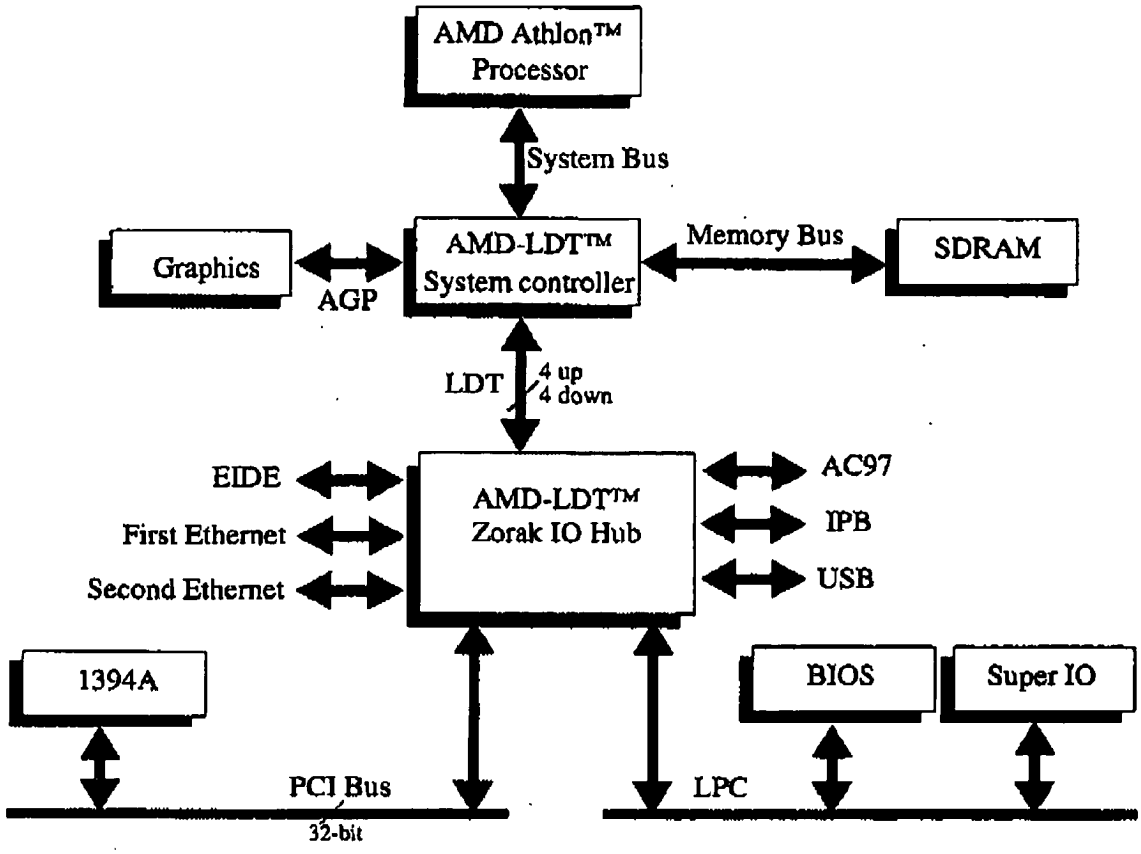
The AMD-LDT™ chipset is a highly integrated system logic solution that delivers enhanced performance and features for the AMD Athlon processor and other AMD Athlon system bus-compatible processors. The AMD-LDT chipset consists of the AMD-LDT™ system controller and the AMD-LDT™ Zorak IO Hub.

The IO Hub includes the following:

- LDT link supports up to 400 megabytes per second of simultaneous input and output bandwidth using 8-bit LDT input and output links running with a 200 MHz (double pumped) LDT clock. .
- Multiple LDT bit widths including 8 bits, 4 bits, and 2 bits (input and output) with a 200 MHz (double pumped) LDT clock supported.
- Version 2.2 compliant PCI bridge. Includes PCI bus arbiter with support for up to eight external devices.
- AC97 soft modem and soft audio interface.
- IPB bus interface to connect to soft DSL modem codecs and other devices.
- Two Ethernet MACs each including an MII interface.
- Embedded controller used in support of the ASF managed desktop specification. Supports internal Ethernet controller or external NIC. Includes SMBus 2.0 interface.

- Two OHCI-based USB hosts supporting a total of 8 USB ports. Each host supports USB specification version 1.1. Each host supports four ports.
- Enhanced IDE controller. Support for a primary and a secondary dual-drive port, PIO modes 1-4, read prefetch and posted write buffers for PIO accesses, bus master IDE, UDMA (through to ATA-100), ATAPI, 2 separate FIFOs for DMA accesses.
- LPC bus to connect peripherals such as super IO and BIOS.
- Support for an external PCI-ISA bridge. PC/PCI and serial IRQ protocol support included.
- Extensive ACPI-compliant, power management logic including highly-programmable C2, C3, power-on-suspend states, suspend to RAM, suspend to disk, throttling, numerous hardware traps, and several system inactivity timers. Includes full SMBus 1.0 interface.
- 32 general purpose IO (GPIO) pins.
- Privacy/security logic; BIOS access control.
- Legacy AT-compatible blocks: dual-8259 interrupt controller; 8254 programmable interval timer; dual-8037 DMA controller (for PC/PCI and LPC bus); legacy support logic including port 61, port 92, gate A20, etc.
- 82093-compatible IOAPIC.
- AT-compatible, year 2000 compliant real-time clock. Includes 512 bytes of CMOS, battery-powered RAM and ACPI-compliant extensions.
- Random number generator.
- 492-pin BGA.
- 2.5-volt core; 3.3-volt output drivers; 5-volt tolerant input buffers.

Figure 1. System Block Diagram



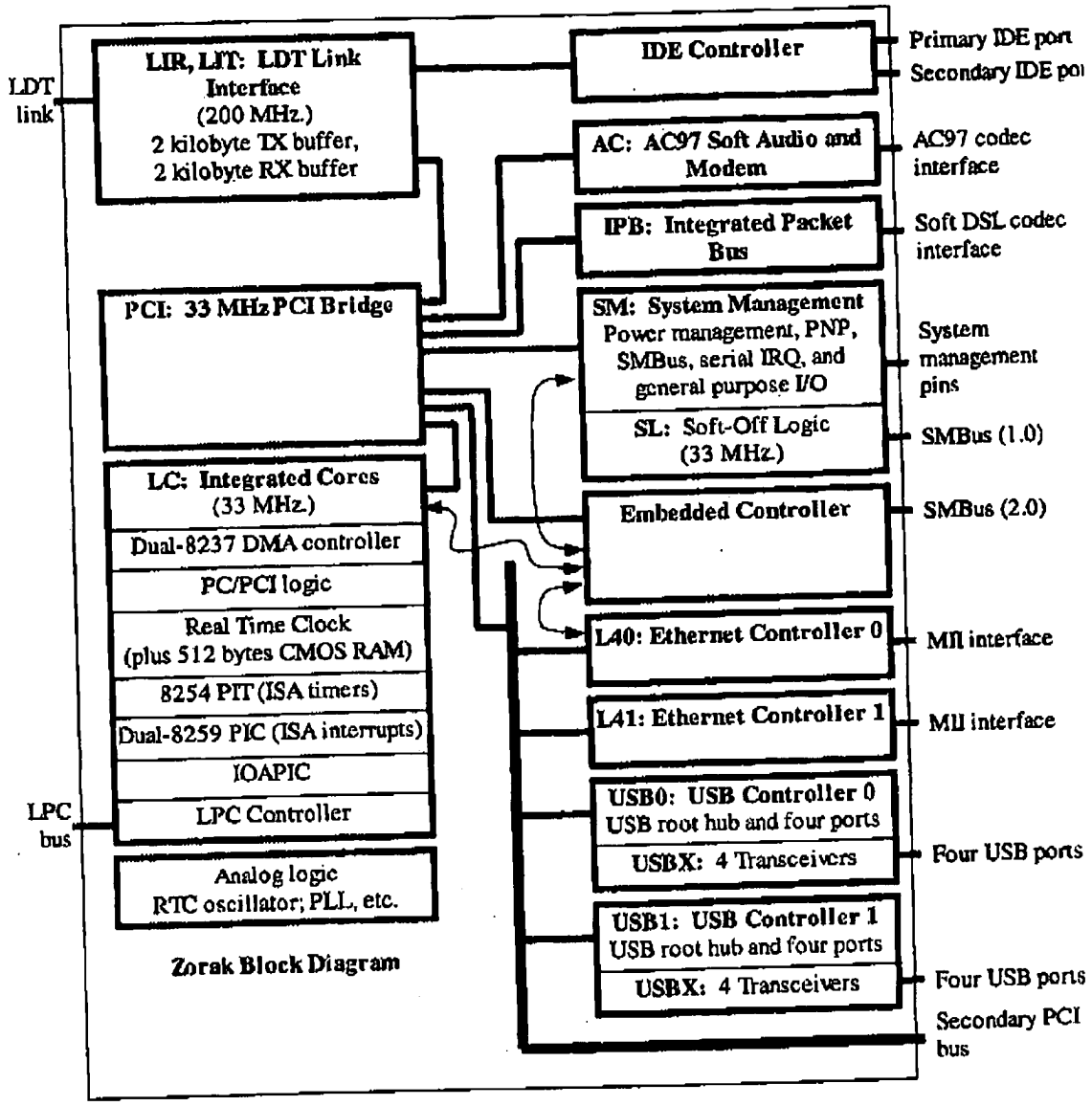


Figure 2. Zorak Block Diagram

4.10 Embedded Controller

Revision History

Revision	Date	Comments	Author
0.0	3/1/00	Under Construction	J Winkler
0.1	3/23/00	Initial Release	J Winkler
0.2	4/4/00	- GPIO section added - watchdog timer reset added to reset section - EC_EN added to LOCK_LC and clock section	J Winkler
0.3	5/9/00	- separate UART block in Figure 4-1 deleted - EC_LOCK_LC_EN added to master access descriptions - master interface interrupt to be configured edge-sensitive - added alternate functions for EC_GPIO[7:5] - 16k bytes of internal program/data RAM	J Winkler
0.4	7/11/00	- HCYC interrupt moved to PFI - boot ROM extended to 512 bytes	J Winkler
Revision Number changed to general MAS Revision			
0.4	9/29/00	- PFI redefined to level-sensitive by Synopsys - no target interface lock after target access from host	J Winkler

Notes to the logic designer are preceeded by a [DesignNote] label and tagged 'AMD Internal' conditional text.

Important specification topics are preceeded by a [SpecAlert] label and tagged 'AMD Internal' conditional text.

Open Issues:

- none :)

4.10.1 Overview

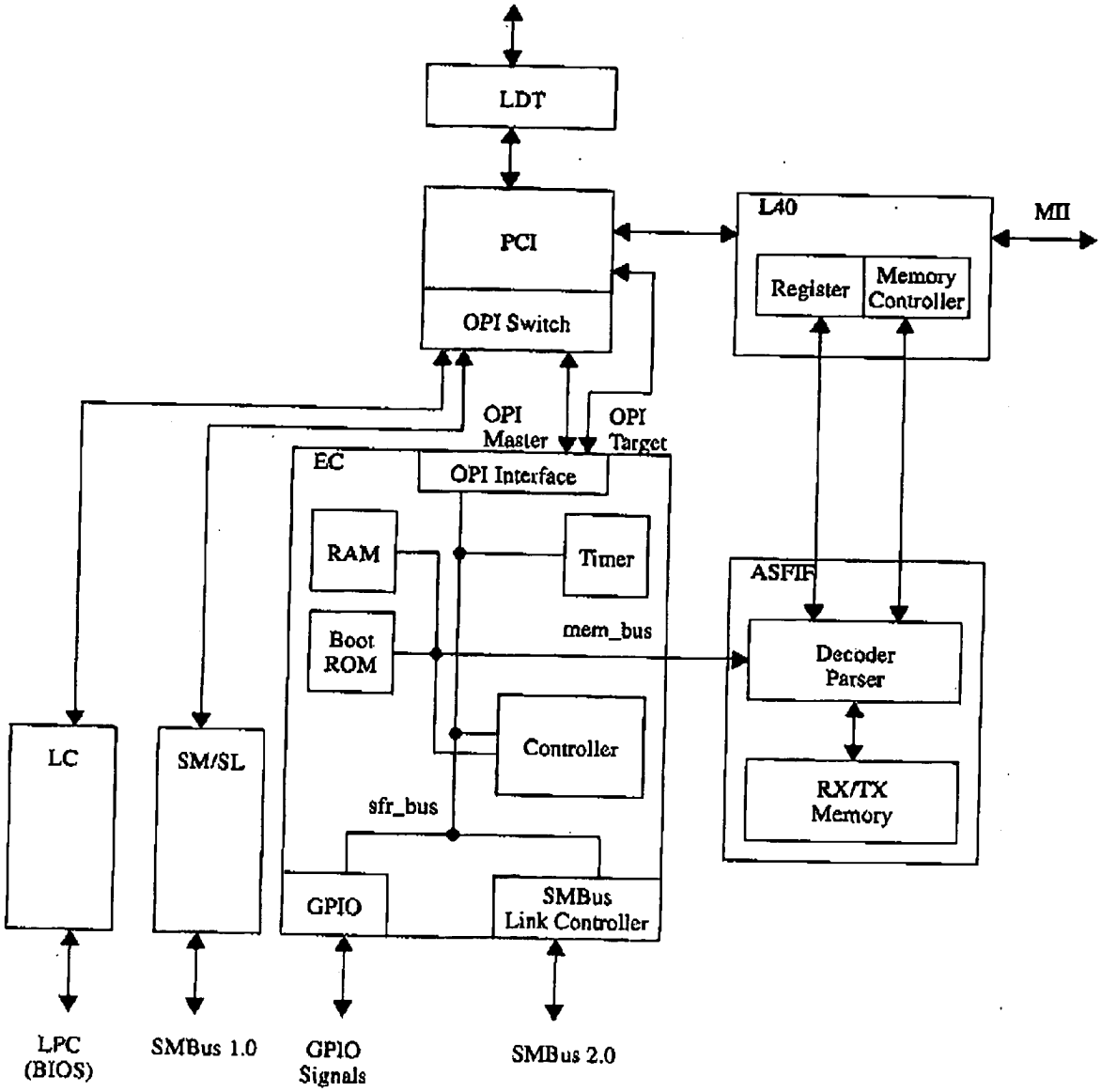
The IC' embedded controller (EC) provides resources to implement ASF related functionlity. Those features comprise interpreting RMCP messages, generating PET messages, and controlling an SMBus 2.0 complient system management bus. The EC features an OPI master and target interface, an ASF interface to the L40 ethernet media access controller and an SMBus 2.0 compliant interface. In addition 8 general purpose signals are provided.

The embedded controller is an 8051 compliant core implementing the following features:

- 4 clocks per instruction cycle
- 2 data pointers
- Stretch memory cycle capability
- 16-bit timers 0 - 2
- Serial port 0 (UART)
- 13 interrupt sources handled by interrupt unit 0 and 1
- 128 bytes of internal RAM
  [DesignNote] Implement as registers. For power reduction latch write accesses with negedge clock and use posedge of iram_we for actual register write.
- 512 bytes of boot ROM (mapped into external RAM address space)
  [DesignNote] To be implemented with "Frank's ROM Generator".
- 16k bytes of program and data RAM (mapped into external RAM address space)

Figure 4-1 shows the block structure of EC.

Figure 4-1 EC Block Structure

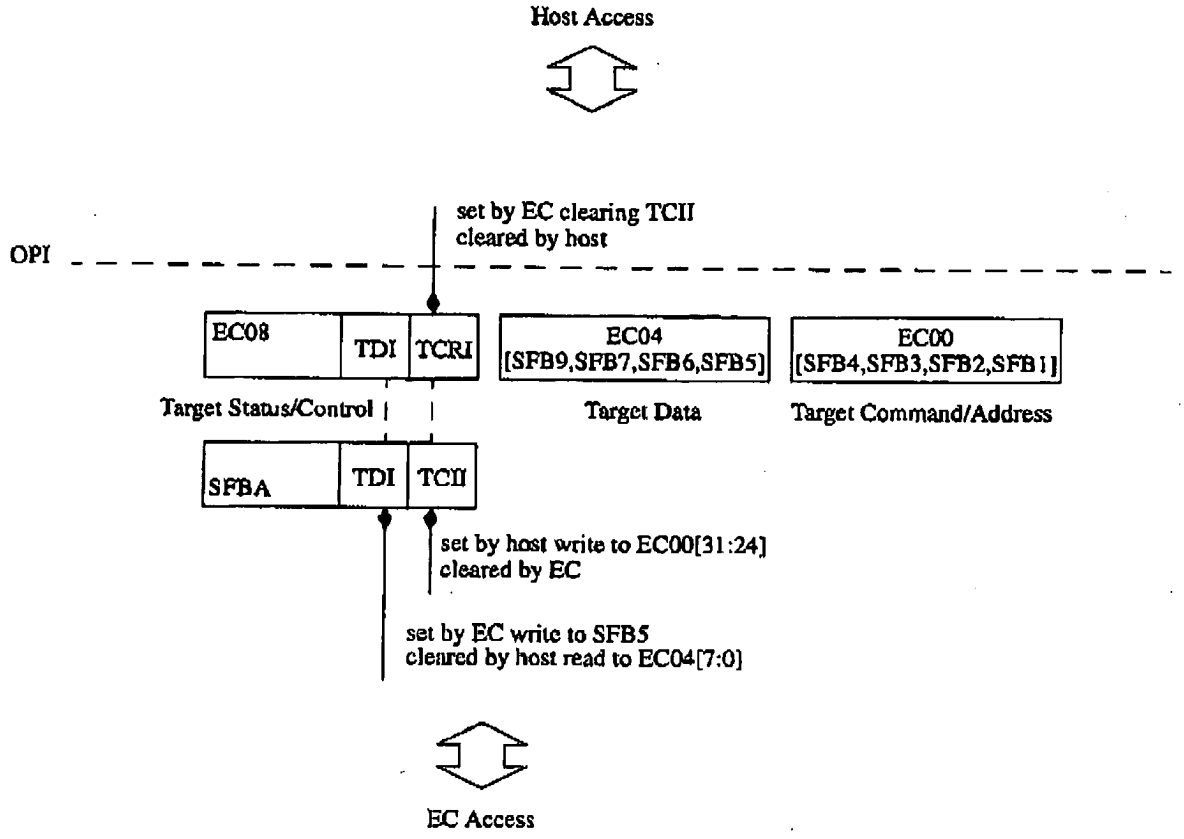


4.10.2 OPI Access

4.10.2.1 Target Accesses

Target accesses to the EC from the host are realized by a command/data port interface incorporating a bi-directional interrupt scheme. Figure 4-2 shows the principal structure of the EC target interface.

Figure 4-2 EC Target Interface



The Target Command/Address (TCA) register is a 32 bit wide register which contains both the command and the address field for the target operation. The width and the interpretation of the bit fields is left to the firmware implementation.

The Target Data (TDAT) register is a 32 bit wide register which transports both read and write data. The implementation comprises two physically different registers, one for write operations and one for read operations. Thus a read to that register will not necessarily return the recently written data.

The Target Status/Control (TSC) register is a 32 bit wide register which provides status information and control of the target interface operations.

The target interface does not support burst transfers.

Write Operation

- The host writes data to TDAT and command/address information to TCA. The write access to the MSB of TCA issues a Target Command Issue Interrupt (TCII) to the EC.
- EC reads TCA and TDAT
- EC clears TCII. By that an optional Target Comand Ready Interrupt (TCRI) can be issued to the host to notify that the command is being processed.

TCRI is a means to notify the host about the completion of the current transaction. After TCRI has been asserted subsequent accesses can be issued by the host without overwriting data in the interface registers.

Read Operation

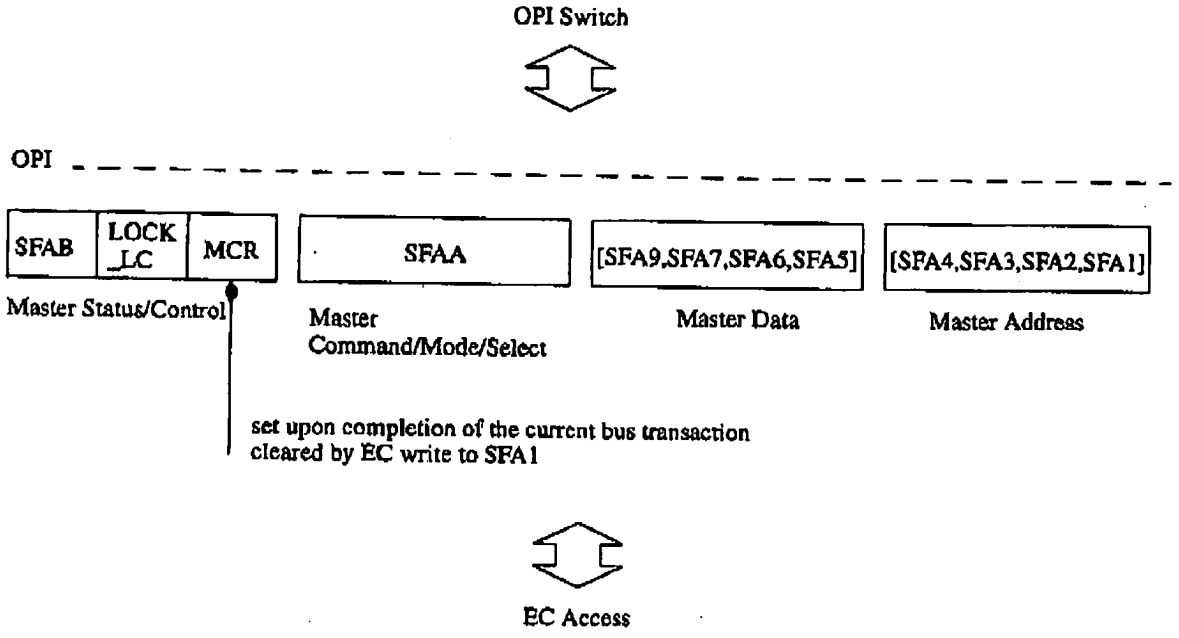
- The host writes command/address information to TCA. A write access to the MSB of TCA issues a TCII to the EC.
- EC reads TCA and processes command.
- EC writes requested data to TDAT. The write access to the LSB of TDAT sets the Target Data Interrupt (TDI) to notify the host about the availability of the requested data.
- EC clears TCII.
- The host reads data from TDAT. The read access to the LSB of TDAT will clear TDI.

TDI is an optional means to notify the host about the availability of the requested data in TDAT. After TDI has been asserted subsequent accesses can be issued by the host without overwriting data in the interface registers.

4.10.2.2 Master Accesses

The EC Master Interface provides access pathes through the OPI interface to the legacy logic (LC) and system management logic (SM). Figure 4-3 shows the principal structure of the EC target interface.

Figure 4-3 EC Master Interface



The Master Address (MADDR) register is a 32 bit wide register which carries the address offset for access to legacy logic (LC) or system management logic (SM). The selection between accesses to legacy logic or system management logic is done by means of the Select bit field in the Master Command/Mode/Select register.

The Master Data (MDAT) register is a 32 bit wide register which transports both read and write data. The implementation comprises two physically different registers, one for write operations and one for read operations. Thus a read to that register will not necessarily return the recently written data.

The Master Command/Mode/Select (MCMS) register is an 8 bit wide register. It provides a command bit field for specifying the OPI command, a mode bit field for specifying the data width mode, and a select bit field for specifying an access to either the legacy logic or the system management logic.

The Master Status/Control (MSC) register is a 8 bit wide register which provides status and control information about the master operations.

Write Operation

- EC checks the MCR bit in MSC to be set.
- EC optionally sets the LOCK_LC bit in MSC. This bit controls the allocation of accesses to the legacy logic for the EC only. When set the access to the legacy logic will be locked after the

current transaction is finished and released only after that bit is cleared. This bit is and'ed with the DevB:3x48[ECLOCKLC] bit, i.e. when DevB:3x48[ECLOCKLC] is set to 0b no locking can occur.

- EC modifies the command, mode, and select bit fields in MCMS if required.
- EC writes data to MDAT.
- EC writes address information to MADDR. Write access to the LSB of MADDR starts the OPI transaction. Write access to the LSB of MADDR will also clear the MCR bit in MSC indicating that an OPI transaction is in progress.
- EC polls for the MCR bit in MSC being set. This will indicate that the master interface is ready for the next master operation. Alternatively the MCR bit may be enabled to issue an interrupt.

Read Operation

- EC checks the MCR bit in MSC to be set.
- EC optionally sets the LOCK_LC bit in MSC. This bit controls the allocation of accesses to the legacy logic for the EC only. When set the access to the legacy logic will be locked after the current transaction is finished and released only after that bit is cleared. This bit is and'ed with the DevB:3x48[ECLOCKLC] bit, i.e. when DevB:3x48[ECLOCKLC] is set to 0b no locking can occur.
- EC modifies the command, mode, and select bit fields in MCMS if required.
- EC writes address information to MADDR. Write access to the LSB of MADDR starts the OPI transaction. Write access to the LSB of MADDR will also clear the MCR bit in MSC indicating that an OPI transaction is in progress.
- EC polls for the MCR bit in MSC being set. This will indicate that the master interface has received the requested data. Alternatively the MCR bit may be enabled to issue an interrupt.
- EC reads the requested data from MDAT.

4.10.3 SMBus Link Controller

The SMBus link controller provides to the EC a register based interface for transmitting and receiving data over an SMBus 2.0 complient interface. For a detailed description of the SMBus interface see the SMBus 2.0 specification.

The SMBus interface consists of two interface signals between the controller and external SMBus devices. Table 4-1 indicates the SMBus signal pins on the IC.

Table 4-1 SMBus Signal Pins

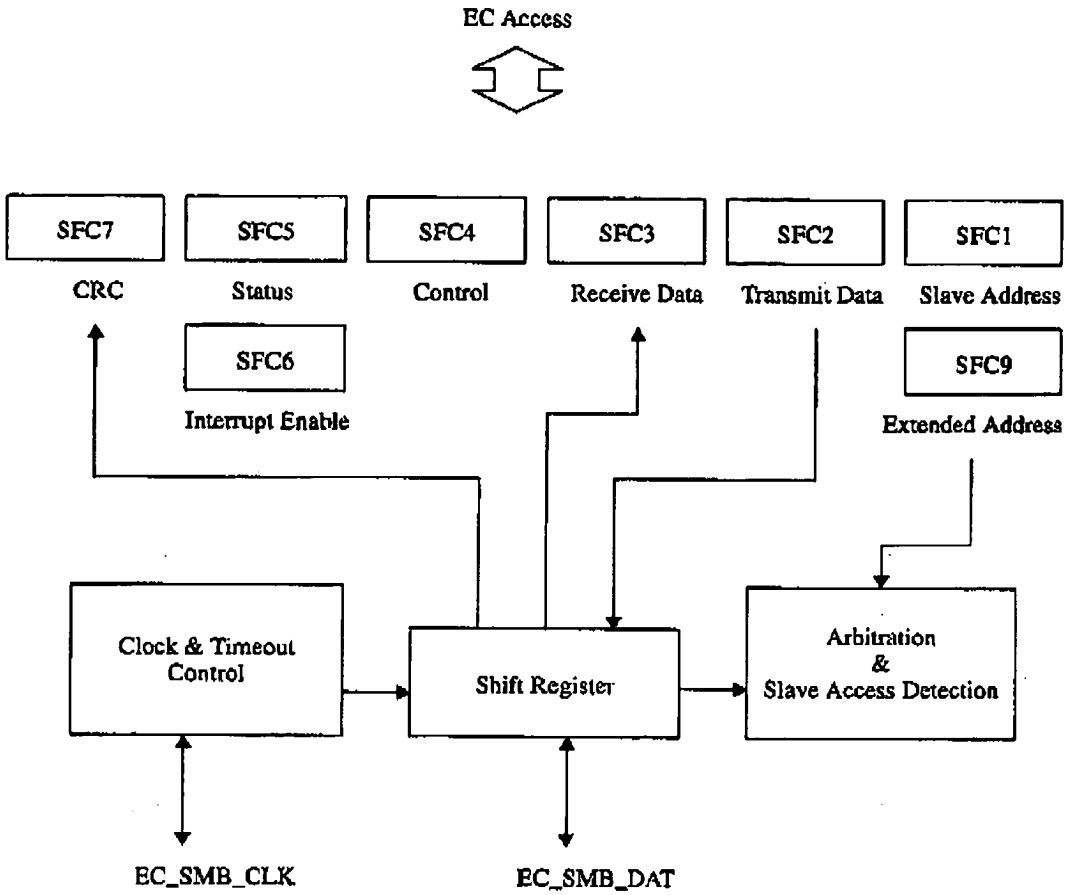
Signal	I/O	Power Plane	Description
EC_SMB_CLK	B	VDD_AUX	SMBus Clock. 100 kHz maximal frequency.

Table 4-1 SMBus Signal Pins

Signal	I/O	Power Plane	Description
EC_SMB_DAT	B	VDD_AUX	SMBus Data.

Figure 4-4 shows the principal structure of the SMBus link controller interface.

Figure 4-4 SMBus Link Controller Interface



4.10.3.3 Transaction Control

The SMBus link controller can act in either master or slave mode. In both modes the SMBus link controller can be either transmitter or receiver.

In master mode the EC initiates an SMBus message. The first transaction of each message is the transmission of a seven bit wide slave address and an attached read/write bit. Before intiating that master transmitter transaction the EC ought to check SFC5[IDLE] to make sure that the SMBus is in idle mode i.e. no transaction is conducted. In any case the SMBus link controller will wait to start the transaction until it detects a bus idle condition. When the bus idle condition is detected the SMBus link controller starts shifting out the data byte to EC_SMB_DAT. In parallel the arbitration control logic checks the data received from EC_SMB_DAT. Once a mismatch between transmitted and received data is detected an arbitration lost condiditon is signaled by setting SFC5[LARB] to 1b. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively, and the SMBus link controller becomes idle.

In slave mode an other SMBus master is sending a slave address to which the IC has to respond with an acknowledge condition. When SFC4[SLCEN] is set to 1b the SMBus link controller will react on incoming data. After a start condition had been detected the arbitration control logic will receive the next 8 bit being transmitted on EC_SMB_DAT. The SMBus link controller will set SFC4[DREQ] and SFC4[SASTA] to indicate that a slave address has been received. When SFC9[SAA] is set to 1b an acknowledge condition will be automatically generated when the received data matches either SFC1 for one of the addresses enabled by SFC9[HOST], SFC9[ALERT], or SFC9[GENERAL]. When SFC9[SAA] is set to 0b no automatic address acknowledge will be generated. In case of an accepted slave address firmware needs to set SFC4[ACK] to 1b which will trigger the generation of an acknowledge condition. When the arbitration control logic detects a stop condition at the end of a slave access SFC5[SASTO] is set.

The SMBus link controller incorporates two timers which control the expiration of the allowed clock low extend times for slave and master devices. The timers are automatically started at the beginning of a SMBus message for slave devices and at the beginning of a byte transaction for master devices. The timers are reset at the end of a SMBus message for slave devices and at the end of a byte transaction for master devices. For a more detailed description of the clock low extend times refer to the SMBus 2.0 specification. Once one of these timers expires the appropriate bit STOUT or MTOUT in SFC5 is set. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively. When the SMBus link controller is in master mode a stop condition is generated automatically and the SMBus link controller becomes idle. When the SMBus link controller is in slave mode it becomes idle.

4.10.3.4 Transaction Trigger

The SMBus link controller has two transaction control modes. In normal transaction control mode the transaction is triggered by valid data in SFC4.

A transmit transaction in automatic transaction control mode is triggered by valid data in SFC2 . Once the link controller detects the trigger condition the data in SFC2 and SFC4 will be shadowed into internal registers and SFC5[DREQ] will be asserted to signal that the data in SFC2 and SFC4 have been invalidated. While the SMBus link controller is using the shadowed data from SFC2 and SFC4 for the current transaction the EC can write data to SFC2 and SFC4 for the next transaction.

A receive transaction in automatic transaction control mode is triggered by invalid data in SFC3 i.e. by providing space for storing received data. Once the link controller detects the trigger condition it writes the received data byte into SFC3, shadows SFC4 into an internal register and asserts SFC5[DREQ] to signal that there are valid data in SFC3 and that the data in SFC4 has been invalidated. While the SMBus controller receives the next byte using the control information provided by the shadowed data from SFC4 the EC can read the data from SFC3 and write data into SFC4 for the next transaction.

4.10.3.5 Master Transmitter Transaction

The data byte to be transmitted in the next transaction has to be provided in SFC2. The control information for controlling the next transaction has to be provided in SFC4.

SLCEN is required to be set to 1b whenever the controller is to accept a transaction. When the byte transaction is the first one of an SMBus message START needs to be set in order to generate a start condition at the first clock of that transaction. When the byte transaction is the last one of an SMBus message STOP needs to be set in order to generate a stop condition at the last clock of that transaction. Generating a stop condition will also cause SFC4[ATC] to be cleared. This will prevent unexpected behavior of the SMBus link controller in case of an immediately incoming slave access. To check for an ACK condition in the acknowledge phase of the transaction the ACK flag needs to be set to 1b. In case of an acknowledge phase mismatch SFC5[AERR] will be set. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively. If SFC4[ASG] is set to 1b a stop condition will be automatically generated and the SMBus link controller becomes idle. Otherwise the SMBus link controller is waiting for the next transaction trigger. When the next transaction is to be triggered by valid data in SFC2 SFC4[ATC] has to be set to 1b. Otherwise the next transaction is triggered by valid data in SFC4.

4.10.3.6 Master Receiver Transaction

Data in SFC3 has to be invalidated in order to provide space for storing the data byte to be received. The control information for controlling the next transaction has to be provided in SFC4.

SLCEN is required to be set to 1b whenever the controller is to accept a transaction. When the byte transaction is the last one of an SMBus message STOP needs to be set in order to generate a stop condition at the last clock of that transaction. Generating a stop condition will also cause SFC4[ATC] to be cleared. This will prevent unexpected behavior of the SMBus link controller in case of an immediately incoming slave access. To generate an ACK condition in the acknowledge phase of the transaction the ACK flag needs to be set to 1b. If the transaction is the last one of an SMBus message

the ACK flag needs to be set to 0b in order to generate a NACK condition flaging the end of the master receiver transaction. When the next transaction is to be triggered by valid data in SFC2 SFC4[ATC] has to be set to 1b. Otherwise the next transaction is triggered by valid data in SFC4.

4.10.3.7 Slave Transmitter Transaction

The data byte to be transmitted in the next transaction has to be provided in SFC2. The control information for controlling the next transaction has to be provided in SFC4.

SLCEN is required to be set to 1b whenever the controller is to accept a transaction. To check for an ACK condition in the acknowledge phase of the transaction the ACK flag needs to be set to 1b. To check for a NACK condition in the acknowledge phase at the end of the transaction the ACK flag needs to be set to 0b. In case of an acknowledge phase mismatch SFC5[AERR] will be set. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively. If SFC4[ASG] is set to 1b a stop condition will be automatically generated and the SMBus link controller becomes idle. Otherwise the SMBus link controller is waiting for the next transaction trigger. When the next transaction is to be triggered by valid data in SFC2 SFC4[ATC] has to be set to 1b. Otherwise the next transaction is triggered by valid data in SFC4.

4.10.3.8 Slave Receiver Transaction

Data in SFC3 has to be invalidated in order to provide space for storing the data byte to be received. The control information for controlling the next transaction has to be provided in SFC4.

SLCEN is required to be set to 1b whenever the controller is to accept a transaction. To generate an ACK condition in the acknowledge phase of the transaction the ACK flag needs to be set to 1b. In case of not acknowledging the received data, e.g. a not supported command, the ACK flag needs to be set to 0b in order to generate a NACK condition. When the next transaction is to be triggered by invalid data in SFC3 SFC4[ATC] has to be set to 1b. Otherwise the next transaction is triggered by valid data in SFC4.

4.10.4 General Purpose IO

The EC provides for 8 dedicated general purpose IO ports accessible at pins EC_GPIO[7:0]. Table 4-2 indicates the EC specific general purpose IO signal pins dedicated to EC.

Table 4-2 General Purpose IO Signal Pins

Signal	I/O	Power Plane	Description
EC_GPIO[7:0]	B	VDD_AUX	General Purpose IO.

The GPIO ports are accessible thorugh register SF90. These ports are controlled by register SFE2 through SFE5.

As alternate function the GPIO pins EC_GPIO[1:0] provide access to the serial port of EC. In addition to that the GPIO pins EC_GPIO[7:5] can be used for PLL testing and auxiliary clock sourcing in PLL testmode and PLL bypass mode, respectively.

4.10.5 Interrupts

Table 4-3 lists the PCI Interrupt sources of EC.

Table 4-3 PCI Interrupt Sources

Interrupt Register Bit	Enable Register Bit	Interrupt
EC08[TFE]	EC08[TFE_IEN]	Target Fatal Error Interrupt
EC08[TDI]	EC08[TDI_IEN]	Target Data Interrupt
EC08[TCRI]	EC08[TCRI_IEN]	Target Command Ready Interrupt

Table 4-4 lists the interrupt sources of the embedded controller itself.

Table 4-4 Embedded Controller Interrupt Sources

Interrupt Register Bit	Enable Register Bit	Interrupt
SF91[4]	SFE8[0]	Heartbeat Interrupt (Timer 4) [DesignNote] DW8051 External Interrupt 2 active high, edge-sensitive
SF88[1]	SFA8[0]	Master Interface Interrupt [DesignNote] DW8051 External Interrupt 0 active low, configured as edge-sensitive
SF88[3]	SFA8[2]	Target Interface Interrupt [DesignNote] DW8051 External Interrupt 1 active low, configured as level-sensitive

Interrupt Register Bit	Enable Register Bit	Interrupt
SF91[5]	SFE8[1]	SMBus Link Controller Interrupt [DesignNote] DW8051 External Interrupt 3 active low, edge sensitive; use enabled toggle flop to make it behave level sensitive
SF91[6]	SFE8[2]	General Purpose IO Interrupt [DesignNote] DW8051 External Interrupt 4 active high, edge-sensitive; use enabled toggle flop to make it behave level sensitive
SF91[7]	SFE8[3]	ASF Interface Interrupt [DesignNote] DW8051 External Interrupt 5 active low, edge-sensitive; use enabled toggle flop to make it behave level sensitive This is the interrupt signal from the ASFIF.
SFD8[4]	SFD8[5]	SMBus 1.0 Controller Interrupt [DesignNote] DW8051 External Power-Fail Interrupt active high, level-sensitive This is the OR over the interrupt sources in PME0.
SF98[0] (RX) SF98[1] (TX)	SFA8[4]	Serial Port 0 Interrupt [DesignNote] DW8051 Serial Port 0 Interrupt configurable by firmware
SF88[5]	SFA8[1]	Timer 0 Interrupt [DesignNote] DW8051 Timer 0 Interrupt configurable by firmware
SF88[7]	SFA8[3]	Timer 1 Interrupt [DesignNote] DW8051 Timer 1 Interrupt configurable by firmware
SFC8[7]	SFA8[5]	Timer 2 Interrupt [DesignNote] DW8051 Timer 2 Interrupt configurable by firmware

The watchdog timer is controlled by registers SFD1 through SFD3. The heartbeat timer is controlled by registers SFD9 tthrough SFDC.

The Target Interface Interrupt comprises several interrupt sub-sources. The according status flags and interrupt enble bits are defined in SFBA.

The ASF Interface Interrupt is signaling the detection of an incoming ASF ethernet frame.

The SMBus Link Controller Interrupt comprises several interrupt sub-sources. The according status flags are defined in SFC5 with interrupt enble bits defined in SFC6.

The General Purpose IO Interrupt comprises several interrupt sub-sources. The according status flags are defined in SFE4 with interrupt enble bits defined in SFE6.

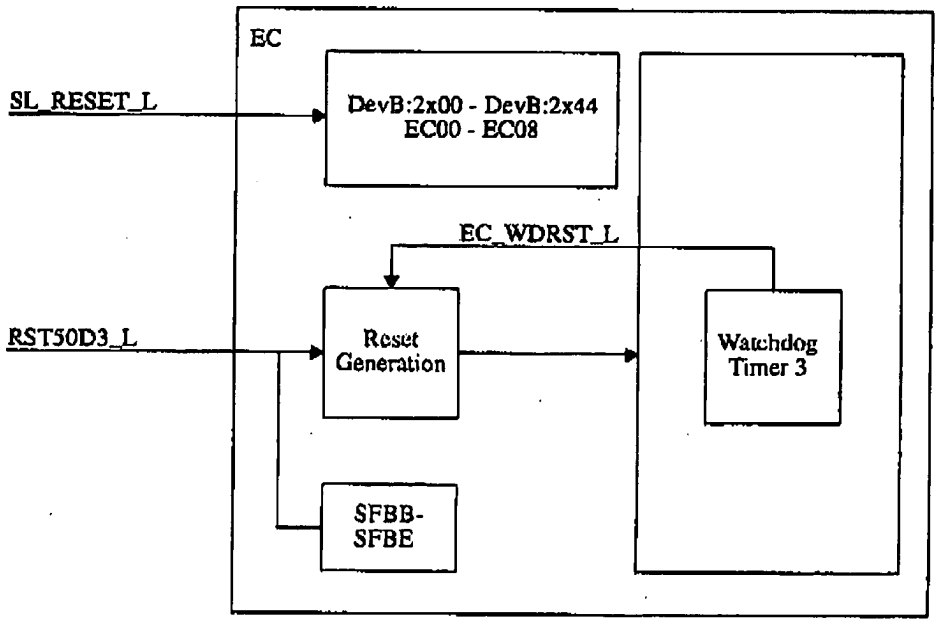
4.10.6 Resets

The registers DevB:2x00 through DevB:2x44 and EC00 through EC08 are reset by assertion of SM_RESET_L. All remaining logic in EC is reset by assertion of RST50D3_L. In addition that logic excluding the registers SFBB through SFBE is also reset by assertion of EC_WDRST_L upon expiration of watchdog timer 3.

Once RST50D3_L or EC_WDRST_L had been asserted reset is hold active for a period of 8 clocks of CLK50D3 after deassertion of these signals.

Figure 4-5 shows the reset structure of EC.

Figure 4-5 Reset Structur



4.10.7 Clocks

The registers DevB:2x00 through DevB:2x44 and EC00 through EC08 as well as all accesses to them are driven by PCLK.

All other logic is driven by CLK50D3. This clock can be disabled by DevB:3x64[EC_EN].

All SMBus operations driven by EC_SMB_CLK are run at a frequency of 100 kHz.

4.10.8 Power Wells

All logic in EC is powered by VDD_AUX.

0	TCRI. Read/Write. Target Command Ready Interrupt Status. The embedded controller sets this bit to 1b by clearing SFBA[TCII]. This bit is cleared by writing 1b. This bit is intended for the purpose of signaling that the embedded controller has processed the data provided through the target command interface.

5.11.3 Embedded Controller Special Function Registers

The special function registers are located in the special function address space of the embedded controller. Access for host operations is provided through the embedded controller command interface.

Table 5-2 shows the register address space for the special function registers located in the Special Function Address space of the embedded controller.

Table 5-2 Special Function Register Address Space

Mnemonic	Register
SF81	Stack Pointer (see 8051 Datasheet)
SF82	Data Pointer Low 0 (see 8051 Datasheet)
SF83	Data Pointer High 0 (see 8051 Datasheet)
SF84	Data Pointer Low 1 (see 8051 Datasheet)
SF85	Data Pointer High 1 (see 8051 Datasheet)
SF86	Data Pointer Select (see 8051 Datasheet)
SF87	Power Control (see 8051 Datasheet)
SF88	Timer/Counter Control (see 8051 Datasheet)
SF89	Timer Mode Control (see 8051 Datasheet)
SF8A	Timer 0 LSB (see 8051 Datasheet)
SF8B	Timer 1 LSB (see 8051 Datasheet)

Mnemonic	Register
SF8C	Timer 0 MSB (see 8051 Datasheet)
SF8D	Timer 1 MSB (see 8051 Datasheet)
SF8E	Clock Control (see 8051 Datasheet)
SF8F	Special Function (see 8051 Datasheet)
SF90	GPIO Port
SF91	External Interrupt Flag (see 8051 Datasheet)
SF92	MPAGE (see 8051 Datasheet)
SF98	Serial Port 0 Control (see 8051 Datasheet)
SF99	Serial Data Buffer 0 (see 8051 Datasheet)
SFA1	Master Address Byte 0
SFA2	Master Address Byte 1
SFA3	Master Address Byte 2
SFA4	Master Address Byte 3
SFA5	Master Data Byte 0
SFA6	Master Data Byte 1
SFA7	Master Data Byte 2
SFA8	Interrupt Enable (see 8051 Datasheet)
SFA9	Master Data Byte 3
SFAA	Master Command/Mode/Select
SFAB	Master Status/Control
SFB1	Target Command/Address Byte 0 (aliased to EC00[7:0])
SFB2	Target Command/Address Byte 1 (aliased to EC00[15:8])

Mnemonic	Register
SFB3	Target Command/Address Byte 2 (aliased to EC00[23:16])
SFB4	Target Command/Address Byte 3 (aliased to EC00[31:24])
SFB5	Target Data Byte 0 (aliased to EC04[7:0])
SFB6	Target Data Byte 1 (aliased to EC04[15:8])
SFB7	Target Data Byte 2 (aliased to EC04[23:16])
SFB8	Interrupt Priority (see 8051 Datasheet)
SFB9	Target Data Byte 3 (aliased to EC04[31:24)
SFBA	Target Status/Control
SFBB	EC Event Byte 0
SFBC	EC Event Byte 1
SFBD	EC Event Byte 2
SFBE	EC Event Byte 3
SFC1	SMBus Slave Address
SFC2	SMBus Transmit Data
SFC3	SMBus Receive Data
SFC4	SMBus Control
SFC5	SMBus Status
SFC6	SMBus Interrupt Enable
SFC7	SMBus CRC
SFC8	Timer 2 Mode Control (see 8051 Datasheet)
SFC9	SMBus Extended Address
SFCA	Timer 2 Reload Capture LSB (see 8051 Datasheet)
SFCB	Timer 2 Reload Capture MSB (see 8051 Datasheet)

Mnemonic	Register
SFCC	Timer 2 LSB (see 8051 Datasheet)
SFCD	Timer 2 MSB (see 8051 Datasheet)
SFCE	SMBus Timeout
SFD0	Program Status Word (see 8051 Datasheet)
SFD1	Timer 3 Byte 0
SFD2	Timer 3 Byte 1
SFD3	Timer 3 Byte 2
SFD4	Timer 3 Byte 3
SFD8	External Interrupt Control (see 8051 Datasheet)
SFD9	Timer 4 Byte 0
SFDA	Timer 4 Byte 1
SFDB	Timer 4 Byte 2
SFDC	Timer 4 Byte 3
SFE0	Accumulator (see 8051 Datasheet)
SFE2	GPIO Control
SFE3	GPIO Interrupt Polarity
SFE4	GPIO Interrupt Status
SFE5	GPIO Interrupt Enable
SFE8	Extended Interrupt Enable (see 8051 Datasheet)
SFF0	B Register (see 8051 Datasheet)
SFF8	Extended Interrupt Priority (see 8051 Datasheet)

The subsequent register naming uses occassionally a suffixed "x" to indicate that those register definitions apply in the same way for the appropriate registers with the suffix number 0 through 3.

GPIO Port
EC_SF_GP **SF90**

Default: FFh Attribute: Read/Write.

Bit	Description
7:0	GP. General Purpose IO Port. These bits reflect the state of the respective general purpose IO pins EC_GPIO[7:0]. A write to pins which are configured as inputs will be stored in the internal output value register but will not affect the read value of this bit.

As alternate function EC_GPIO[1:0] provides access to the serial port of EC. EC_GPIO[1] transmits the serial port data in serial port mode 1, 2, 3. For that SFE6[1] and SFE2[1] need to be set to 1b. EC_GPIO[0] receives the serial port data in serial port mode 1, 2, 3. For that SFE2[0] needs to be set to 0b. Serial port mode 0 is not supported. For normal usage of EC_GPIO[1:0] SFE6[1] needs to be set to 0b and the receive operation of the embedded controller needs to be disabled to avoid interrupt generation from changing signals at EC_GPIO[0]. [DesignNote] ec_gpio[1] <> txd0, ec_gpio[0] <>rxd0_in

As an alternate function EC_GPIO[6] provides access to the lock output signal of S3PLL and EC_GPIO[5] to the clock output signal of S3PLL when DevB:3x48[RPTEST] is set to 1b (RNG and PLL test mode). [DesignNote] Use sm_c3a48_rptest to set those pins to output and to mux out the signals.

As alternate function EC_GPIO[7] can be used to provide an auxiliary clock to be used instead of the S3PLL clock outputs when DevB:3x48[PLLBPM] is set to 1b (PLL bypass mode).

Note: The General Purpose IO control registers are located at SFE2 thorugh SFE5.

Master Address Byte 0, 1, 2, 3
EC_SF_MADDRx **SFA1, SFA2, SFA3, SFA4**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	Master Address Byte x.

SFA1 is the least significant byte and SFA4 is the most significant byte of a composed master address field.

Master Data Byte 0, 1, 2, 3
EC_SF_MDATx **SFA5, SFA6, SFA7, SFA9**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	Master Data Byte x.

SFA5 is the least significant byte and SFA9 is the most significant byte of a composed master data field.

[DesignNote] These registers are aliased to two different sets of registers, one for write operations and one for read operations. Thus a read to that address might not return the data recently written to that address.

Master Command/Mode/Select EC_SF_MCMS — SFAA

Default: 00h Attribute: Read/Write.

Bit	Description
7:4	**CMD.** Master Command. OPI bus command bits as provided to the bus interface unit. The master interface supports IO read, IO write, memory read, posted and non-posted memory write, configuration read, and configuration write commands as defined by the OLI/OPI command table.
3:2	**MODE.** OPI data width mode select. 0h - Byte mode. Only SFA5 is enabled for data transfers on the OPI. SFA1 represents a byte aligned address. 1h - Word mode. Only SFA5 and SFA6 are enabled for data transfers on the OPI. SFA1 represents a word aligned address. 3h - Double word mode. SFA5, SFA6, SFA7, and SFA9 are enabled for data transfers on the OPI. SFA1 represents a double word aligned address. 2h - Reserved.
1:0	**SELECT.** Target select. These bits select the target for the subsequent commands at the master interface. 0h - LC 1h - SM 2h-3h - Reserved.

Depending on the master mode and master address the respective byte enables for the transfer over OPI are appropriately generated.

Master Status/Control EC_SF_MSC — SFAB

Default: 03h Attribute: See description.

Bit	Description
7:6	**Reserved.**
5	**MCR_EN.** Master Command Ready Interrupt Enable. Read/Write. Set to 1b enables an interrupt caused by a 1b in SFAB[MCR].
4:2	**Reserved.**
1	**MCR.** Master Command Ready. Read only. This bit indicates the completion of a master command operation. It is set upon completion of the current OPI master transaction. It is cleared by an embedded controller write acces to SFA1 upon start of a new master command operation.

0	**LOCK_LC.** Lock LC. Read/Write. This bit controls the allocation of LC for embedded controller accesses only. This bit is and'ed with the DevB:3x48[ECLOCKLC] bit, i.e. when DevB:3x48[ECLOCKLC] is set to 0b no locking can occur. 1b - locks LC after the current bus transfer is completed 0b - releases LC after the current bus transfer is completed

The interrupt caused by SFAB is routed to SF88[1] and can be enabled by SFA8[0]. [DesignNote] See the DW8051 Datasheet.

Target Command/Address Byte 0, 1, 2, 3
EC_SF_TCAx

SFB1, SFB2, SFB3, SFB4

Default: 00h Attribute: Read only.

Bit	Description
7:0	Target Command/Address Byte x.

These registers are aliased to EC00 by [SFB4, SFB3, SFB2, SFB1].

Target Data Byte 0, 1, 2, 3
EC_SF_TDATx

SFB5, SFB6, SFB7, SFB9

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	Target Data Byte x.

These registers are aliased to EC04 by [SFB9, SFB7, SFB6, SFB5].

Target Status/Control
EC_SF_TSC

SFBA

Default: 00h Attribute: See description.

Bit	Description
7	**PD.** Read only. Power Down Status. This bit is set to 1b in case VDD_CORE is not powered or/and SL_RESET_L is asserted. In that system state accesses over the OPI are not possible. To wake up the system EC has to write 1b to SFBB[WUE]. This bit remains set as a flag until the first read access to this bit location. After that first read access this bit follows the power state of VDD_CORE i.e. is deasserted at deassertion of SL_RESET_L. [DesignNote] This bit is derived from "sl_ec_goto_s345 | ~sm_reset_l".
6	**TFE.** Read/Write. Target Fatal Error. This bit is set to 1b by EC to signal a fatal error condition to the host. Before setting this bit EC can place an appropriate error code in EC04 which can be evaluated by the host. This bit is cleared when the host clears EC08[TFE].
5	**TDI.** Read only. Target Data Status. This bit is set to 1b by a EC write access to SFB5 when valid data are provided in case of a target read operation. This bit is cleared by a host read acces to EC04[7:0].
4	**TCII.** Read/Write. Target Command Issue Status. This bit is set by a host write access to EC00[31:24]. Setting this bit will also lock the target command interface from further host accesses. This bit is cleared by writing 1b to this bit. Clearing this bit will also release the target command interface for further host accesses.
3	**PD_EN.** Read/Write. Power Down Interrupt Enable. Set to 1b enables an interrupt to the embedded controller caused by an 1b in PD.
2:1	**Reserved.**

0	**TCII_EN.** Read/Write. Target Command Interrupt Enable. Set to 1b enables an interrupt to the embedded controller caused by an 1b in TCII.

The interrupt caused by SFBA is routed to SF88[3] and can be enabled by SFA8[2]. [DesignNote] See the DW8051 Datasheet.

EC Flag Byte 0
EC_SF_TECFx — SFBB

Default: 00h Attribute: Read/Write.

Bit	Description
7:2	**ECF.** EC Flags. These bits are intended to reflect certain operating states and events of the EC firmware. The definition is left to the firmware.
1	**DBGRAM.** External Debug RAM. This bit determines the address mapping of the upper 32k bytes of the EC address space when DevB:3x48[ECMEXT] is set to 1b. When set to 1b the upper 32k bytes of the EC address space are mapped to the EC debug interface. When set to 0b the upper 32k bytes of the EC address space are mapped to the internal program/data RAM.
0	**WUE.** Wake Up Event. Writing 1b to this bit position will cause PM20[EC_STS] to be set. Reads will always return 0b in this bit position. [DesignNote] This bit is routed to SL. When written 1b this bit has to generate a high pulse for one CLK50D3 cycle. This drives ec_set_pm20_ec_sts to SL.

EC Flag Byte 1, 2, 3
EC_SF_TECFx — SFBC, SFBD, SFBE

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	EC Flag Byte x. These bits are intended to reflect certain operating states and events of the EC firmware. The definition is left to the firmware.

SMBus Slave Address
EC_SF_SSADDR — SFC1

Default: 62h Attribute: Read/Write.

Bit	Description
7:1	**SSA.** This address is used to identify the device on the SMBus. It is assigned during the address resolution protocol procedure. This register defaults to the Device Default Address 1100_001b.
0	**Reserved.**

SMBus Transmit Data
EC_SF_STDAT — SFC2

Default: 00h Attribute: Read/Write.

Bit	Description

7:0	**STD.** Data to be transmitted over the SMBus SMBDTA pin. A write acces to this register validates the data. The data is invalidated when read by the SMBus link controller into the internal shift register. Data are always shifted from right to left i.e. the first bit shifted out onto EC_SMB_DAT is bit 7 and the last one bit 0. When transmitting data with SFC4[ATC] set to 1b valid data will trigger the next transaction.

SMBus Receive Data
EC_SF_SRDAT

SFC3

Default: 00h Attribute: Read only.

Bit	Description
7:0	**SRD.** Data received from the SMBus SMBDTA pin. A read access to this register invalidates the data. The data is validated when written by the SMBus link controller from the internal shift register. Data are always shifted from right to left i.e. the first bit shifted in from EC_SMB_DAT is bit 7 and the last one bit 0. When receiving data with SFC4[ATC] set to 1b invalid data will trigger the next transaction.

SMBus Control
EC_SF_SCTRL

SFC4

Default: 00h Attribute: Read/Write.

Bit	Description
7	**Reserved.**
6	**ABORT.** Abort SMBus Link Transaction. When set to 1b SFC4[ATC] will be cleared, SFC2 and SFC4 will be invalidated and SFC3 validated, respectively, and the SMBus link controller becomes idle.
5	**ASG.** Automatic Stop Generation. When set to 1b a stop condition will be automatically generated when SFC5[AERR] is set to 1b and the SMBus link controller is in master mode.
4	**ATC.** Automatic Transaction Control. When set to 1b the automatic transaction control is turned on. In that mode the start of a transmit transaction is triggered by valid data in SFC2 and the start of a receive transaction is triggered by invalid data in SFC3. This transaction is then controlled by the current settings in SFC4. When set to 0b normal transaction control is turned on. In that mode the start of a transmit or receive transaction is triggered by valid data in SFC4.
3	**ACK.** Acknowledge. 0 - Generate or check NACK in the next transmit or receive transaction. 1 - Generate or check ACK in the next transmit or receive transaction.
2	**STOP.** Stop Condition. When set to 1b a stop condition is generated after the next transaction. Generating a stop condition will also cause SFC4[ATC] to be cleared. This will prevent unexpected behavior of the SMBus link controller in case of an immediately incoming slave access.
1	**START.** Start Condition. When set to 1b a start condition is generated before the next transaction.
0	**SLCEN.** SMBus Link Controller Enable. When set to 1b the SMBus link controller is enabled. When set to 0b the SMBus link controller is disabled and all signals on EC_SMB_CLK and EC_SMB_DAT will be discarded. Read and write accesses to the SMBus link controller registers are accepted but will not cause any actions.

A write access to SFC4 validates the data in the register. The data is invalidated when read by the SMBus link controller to the internal shadow register. In normal transaction control mode valid data in SFC4 will trigger the next transaction.

SMBus Status
EC_SF_SSTAT

SFC5

Default: 00h Attribute: Read/Write.

Bit	Description
7	**DREQ.** Data Request. This bit is set to 1b when valid data in SFC2 is required for transmission or when data has to be invalidated by reading from SFC3 for receiption. This bit is cleared by a write access to SFC2 or SFC4 or a read access to SFC3.
6	**IDLE.** Bus Idle. This bit is set to 1b while the bus is idle.
5	**AERR.** Acknowledge Error. This bit is set to 1b when an acknowledge mismatch between the value defined in SFC4[ACK] and the state of EC_SMB_DAT in the ACK phase is detected. In that case SFC4[ATC] will be cleared and SFC2 will be invalidated. If the SMBus link controller is in master mode and SFC4[ASG] is set to 1b a stop condition will be automatically generated and the SMBus link controller becomes idle. If the SMBus link controller is in master mode and SFC4[ASG] is not set the SMBus link controller is waiting for the next transaction trigger. If the SMBus link controller is in slave mode it becomes idle. This bit is cleared by writing 1b to this position.
4	**Reserved.**
3	**TOUT.** Timeout. This bit is set to 1b when at least one of the timeout conditions specified in SFCE[2:0] occured. This bit is cleared when all bits SFCE[2:0] are cleared.
2	**SASTO.** Slave Access Stop. This bit is set to 1b upon receiption of a stop condition on the SMBus after a slave transaction. Setting this bit will also causeSFC4[ATC] to be cleared. This will prevent unexpected behavior of the SMBus link controller in case of an immediately incoming slave access. This bit is cleared by writing 1b to this position.
1	**SASTA.** Slave Access Start. When SFC9[SAA] is set to 0b this bit is set to 1b upon receiption of a slave address after a detected start condition. When SFC9[SAA] is set to 1b this bit is set to 1b only when the received slave address matches either SFC1 or one of the slave addresses enabled by SFC9[HOST], SFC9[ALERT], or SFC9[GENERAL]. This bit is cleared by writing 1b to this position.
0	**LARB.** Lost Arbitration. This bit is set to 1b when in master mode arbitration is lost. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively, and the SMBus link controller becomes idle. This bit is cleared by writing 1b to this position.

SMBus Interrupt Enable EC_SF_SIEN — SFC6

Default: 00h Attribute: Read/Write.

Bit	Description
7	**DREQ_IEN.** Command Done Interrupt Enable. Set to 1b enables an interrupt caused by an 1b in SFC5[DREQ].
6	**Reserved.**
5	**AERR_IEN.** Acknowledge Error Interrupt Enable. Set to 1b enables an interrupt caused by an 1b in SFC5[AERR].
4	**Reserved.**
3	**TOUT_IEN.** Timeout Interrupt Enable. Set to 1b enables an interrupt caused by an 1b in SFC5[TOUT].
2	**SASTO_IEN.** Stop Slave Access Interrupt Enable. Set to 1b enables an interrupt caused by an 1b in SFC5[SASTO].
1	**SASTA_IEN.** Start Slave Access Interrupt Enable. Set to 1b enables an interrupt caused by an 1b in SFC5[SASTA].
0	**LARB_IEN.** Arbitration Interrupt Enable. Set to 1b enables an interrupt caused by an 1b in SFC5[LARB].

The interrupt caused by SFC5 in conjunction with SFC6 is routed to SF91[5] and can be enabled by SFE8[1]. [DesignNote] See the DW8051 Datasheet.

SMBus CRC
EC_SF_SCRC

SFC7

Default: 00h Attribute: Read only.

Bit	Description
7:0	**CRC.** CRC Value. Contains the calculated CRC8 Packet Error Code for both transmit and receive operations. The CRC value is valid with the occurence of SFC5[DREQ]. This register is initialized to its default value upon the occurence of a start condition on the SMBus.

SMBus Extended Addresses
EC_SF_SXADDR

SFC9

Default: 00h Attribute: Read/Write

Bit	Description
7	**SAA.** Slave Address Acknowledge. When set to 1b the reception of a slave address matching SFC1 or matching one of the enabled extended addresses in SFC9 will be automatically acknowledged with an ACK condition on the SMBus. Upon receiption SFC5[SASTA] will be set.
6:4	**Reserved.**
3	**GENERAL.** General Call Address Enable. When set to 1b the General Call Address 0000_000b is enabled for slave address auto acknowledge. Slave address auto acknowledge is controlled by SFC9[SAA].
2	**DEFAULT.** Device Default Address Enable. When set to 1b the Device Default Address 1100_001b is enabled for slave address auto acknowledge. Slave address auto acknowledge is controlled by SFC9[SAA].
1	**ALERT.** Alerst Response Address Enable. When set to 1b the Alert Response Address 0001_100b is enabled for slave address auto acknowledge. Slave address auto acknowledge is controlled by SFC9[SAA].
0	**HOST.** Host Address Enable. When set to 1b the Host Address 0001_000b is enabled for slave address auto acknowledge. Slave address auto acknowledge is controlled by SFC9[SAA].

SMBus Timeout
EC_SF_STOUT

SFCE

Default: 00h Attribute: Read only.

Bit	Description
7:3	**Reserved.**
2	**STOUT.** Slave Extend Timeout. This bit is set to 1b when a SMBus slave extend timeout occurs. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively. When the SMBus link controller is in slave mode it becomes idle. When the SMBus link controller is in master mode no slave extend timeouts are generated. This bit is cleared by writing 1b to this position.
1	**MTOUT.** Master Extend Timeout. This bit is set to 1b when the SMBus link controller is in master mode and a SMBus master extend timeout occurs. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively. A stop condition is generated automatically and the SMBus link controller becomes idle. When the SMBus link controller is in slave mode no master extend timeouts are generated. This bit is cleared by writing 1b to this position.
0	**CTOUT.** Clock Low Timeout. This bit is set to 1b when a SMBus clock low timout occurs. In that case SFC4[ATC] will be cleared, SFC2 will be invalidated and SFC3 validated, respectively. When the SMBus link controller is in master mode a stop condition is generated automatically and the SMBus link controller becomes idle. When the SMBus link controller is in slave mode it becomes idle. This bit is cleared by writing 1b to this position.

Timer 3 Byte 0 EC_SF_TIM3B0 — SFD1

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM3B0.** Byte 0 of Timer 3.

Timer 3 is a 32 bit wide watchdog timer. The reload value for the timer is defined in SFD1 through SFD4. SFD1 is the least significant byte and SFD4 is the most significant byte of the reload value. After reset the timer value is 0h and the timer is stopped. Write access to SFD4 loads and starts the timer. After start the timer counts from the loaded value downto zero. Any subsequent write to SFD4 causes the timer to be reloaded. At the transition from 1h to 0h the timer asserts EC_WDRST_L. Once started the timer can not be disabled. Writing a reload value of 0h will be ignored. The timer is clocked by CLK50D3 (50 MHz) thus covering a time period from 20 ns up to about 85 s.

Timer 3 Byte 1 EC_SF_TIM3B1 — SFD2

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM3B1.** Byte 1of Timer 3.

Timer 3 Byte 2 EC_SF_TIM3B2 — SFD3

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM3B2.** Byte 2of Timer 3.

Timer 3 Byte 3 EC_SF_TIM3B3 — SFD4

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM3B3.** Byte 3 of Timer 3. Write Access to this register starts the timer.

Timer 4 Byte 0 EC_SF_TIM4B0 — SFD9

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM4B0.** Byte 0 of Timer 4.

Timer 4 is a 32 bit wide heartbeat timer. The reload value for the timer is defined in SFD9 through SFDC. SFD9 is the least significant byte and SFDC is the most significant byte of the reload value. After reset the timer value is 0h and the timer is stopped. Write access to SFDC loads and starts the timer. After start the timer counts from the loaded value downto zero. Any subsequent write to SFDC causes the timer to be reloaded. At the transition from 1h to 0h the timer issues an interrupt. This interrupt is routed to SF91[4] and can be enabled by SFE8[0]. [DesignNote] See the DW8051 Datasheet. Once started the timer can not be disabled. The minimal reload value is 8h. For reload values less then 8h interrupt detection is not defined. The timer is clocked by CLK50D3 (50 MHz) thus covering a time period from 160 ns up to about 85 s.

Timer 4 Byte 1
EC_SF_TIM4B1 **SFDA**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM4B1.** Byte 1of Timer 4.

Timer 4 Byte 2
EC_SF_TIM4B2 **SFDB**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM4B2.** Byte 2 of Timer 4.

Timer 4 Byte 3
EC_SF_TIM4B3 **SFDC**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**TIM4B3.** Byte 3 of Timer 4. Write access to this register starts the timer.

GPIO Control
EC_SF_GC **SFE2**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	GC. General Purpose IO Control. These bits control the direction of the respective general purpose IO pins. 1 - pin is output 0 - pin is input

GPIO Interrupt Polarity
EC_SF_GIP **SFE3**

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**GIP.** General Purpose IO Interrupt Polarity. These bits control the polarity of the edge which will set the appropriate interrupt status bits in SFE4. 0 - falling edge 1 - rising edge

GPIO Interrupt Status
EC_SF_GIS — SFE4

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**GIS.** General Purpose IO Interrupt Status. A bit position set to 1b reflects an edge-sensetive event detected in accordance to the edge polarity as set in the respective bit position in SFE3. These bits are cleared by writing a 1b to the respective position.

GPIO Interrupt Enable
EC_SF_GIE — SFE5

Default: 00h Attribute: Read/Write.

Bit	Description
7:0	**GIE.** General Purpose Interrupt Enable. A bit position set to 1b enables an interrupt caused by an 1b in the respective bit position in SFE4.

The interrupt caused by SFE4 in conjunction with SFE5 is routed to SF91[6] and can be enabled by SFE8[2]. [DesignNote] See the DW8051 Datasheet.

GPIO Alternate Function
EC_SF_GAF — SFE6

Default: 00h Attribute: Read/Write.

Bit	Description
7:2	**Reserved.**
1	**SPTXEN.** Serial port transmit enable. When set to 1b the serial port transmit signal of the embedded controller is muxed to the EC_GPIO[1] output pin. When set to 0b the value of SF90[1] is muxed to the EC_GPIO[1] output pin.
0	**Reserved.**

5.11.4 Embedded Controller Memory Space

The embedded controller is able to operate in 3 different memory address map modes. These modes are defined by DevB:3x48[ECMEXT] and SFBB[DBGRAM]. DevB:3x48[ECMEXT] controls the